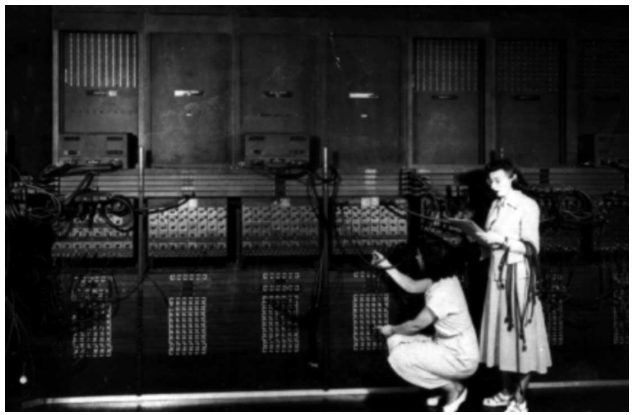
Programming the ENIAC before its rewiring.

The case of the Lehmers' program.

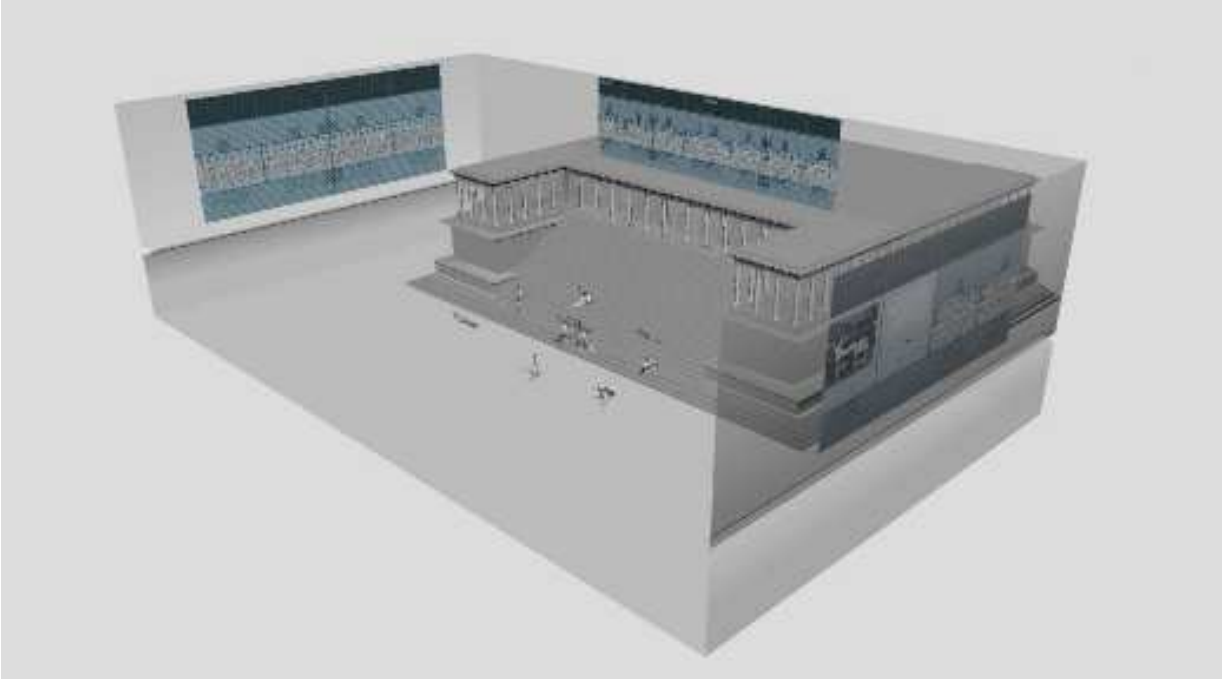


L. De Mol[1] and M. Bullynck[2]

[1] Universiteit Gent, elizabeth.demol@ugent.be

[2] Paris 8, maarten.bullynck@kuttaka.org

Thanks to...



...Martin Carlé and Joulia Strauss and their ENIAC NOMOI project.

Motivation

⇒ ***Historically*:** Understanding the first electronic basically general-purpose US computer and problems it gave rise to. Programming hardware. Start of the necessity to split-up between hardware/software. Significance of reconstructions. How to program a non-logical "behemoth"?

⇒ ***Philosophically*:** Understanding the earliest forms of man-computer communications

"*...we cannot fully understand our own conceptual scheme without plumbing its historical roots...*" (Judson Webb, 1980)

Introduction

1. General historical background
2. How a number-theorist got involved with computers
3. A quick tour through the ENIAC
4. Lehmer's ENIAC program
5. Discussion

General Historical Background

General Historical Background.

- ENIAC, The Electronic(!) Numerical Integrator And Computer
- Initial idea to build a large computer using vacuum tubes: Mauchly who wanted to predict the weather.
- In 1941, Mauchly met Presper J. Eckert at the Moore School at Penn University. Eckert "*was willing and agreeable to talk about the possibility of electronic computers [...] Nobody else really wanted to give it a second thought*" [Mauchly, 1970].

$\Rightarrow$ Formal proposal to the Navy Ordnance for building an electronic computer (mainly to compute firing tables). Eckert and Mauchly started building the ENIAC in 1943.

General Historical Background (continued)

- ENIAC unveiled to the public on February 15, 1946
- 18.000 vacuum tubes; 1.500 relays and 40 panels to form 30 units; mainly, decentralized control system
- Local programming method: "The ENIAC was a son-of-a-bitch to program" (Adèle Goldstine)
- Initially the ENIAC was a highly parallel machine, until it was rewired in 1948:

> *"The original "direct programming" recabling method can best be described as analogous to the design and development of a special-purpose computer out of ENIAC component parts for each new application [...] Anyone now doing research in parallel computing might take a look at ENIAC during this first time period, for indeed ENIAC was a parallel computer with all of the problems and opportunities this entails." [Fritz, 1994]*

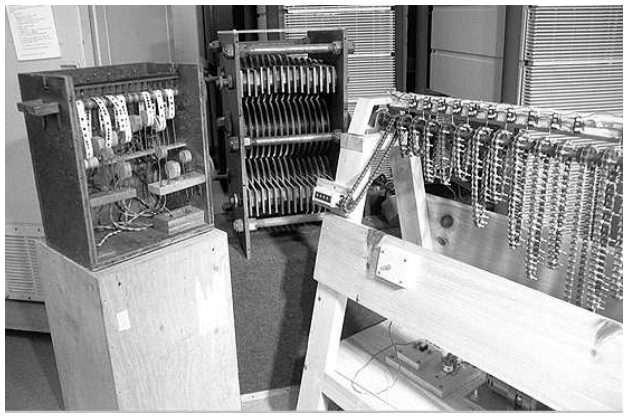
General Historical Background (continued)

- The Ballistic Research Laboratories (Aberdeen Proving Ground) had "assembled a 'Computations Committee' to prepare for utilizing the machine after its completion" [Alt, 1972], and the ENIAC was extensively test-run during its first months.
- The members:
  * Leland B. Cunningham (an astronomer)
  * Haskell B. Curry (a logician)
  * **Derrick H. Lehmer (a number theorist)**

How a number-theorist got involved with computers...

How the Lehmers got involved with Computers.

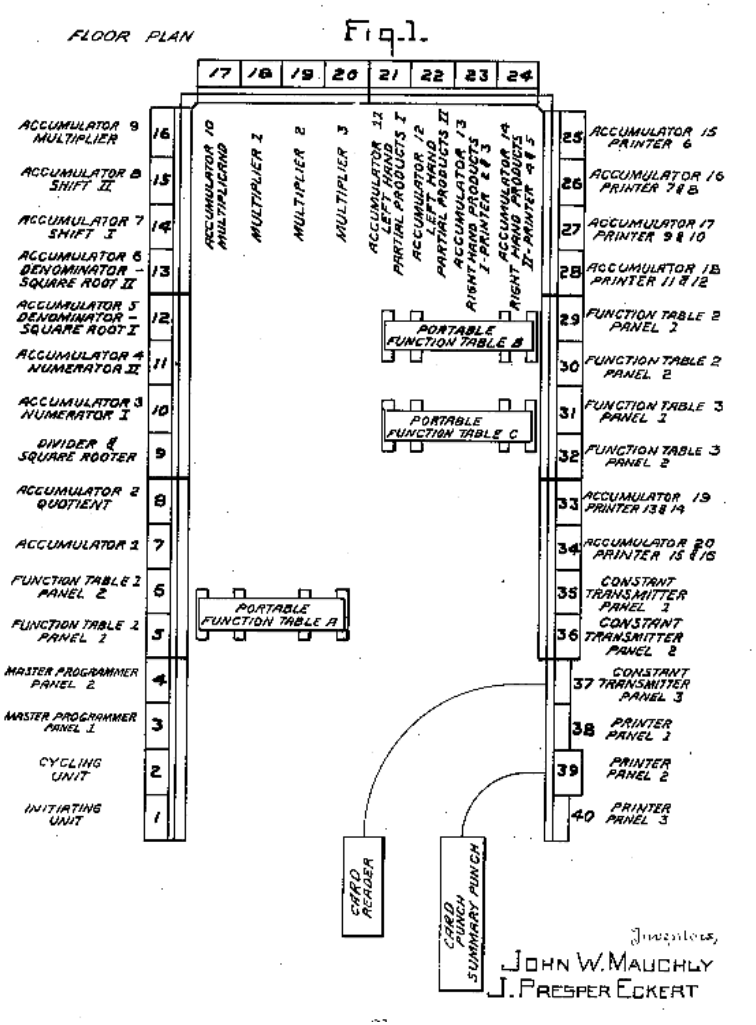
"My father did many things to make me realize at an early age that mathematics, and especially number theory, is an experimental science. If one examines the collected works of Euler, Gauss, Legendre, to name but three, one finds them shamelessly and laboriously computing examples of empirical discoveries. Often these efforts led to the establishment of important theorems. Some of these discoveries remain to this day without logical links to Peano's axioms. [...] We should regard the digital computer system as an instrument to assist the exploratory mind of the number theorist in investigating the global and local properties of this universe, the natural numbers and their algebraic expansions." [Lehmer, 1974]



"I spent [...] two days [...] walking around in the red canyons and exploring the paleontology and archeology of the region [...] On the floor of the canyon are little postholes, and if you investigate one of these you will find a whole little world of its own, living, until it dries out of course, in this very restricted environment. That's the nature of the material I am presenting here. It is really arcane, exotic, and also ancient. We are discussing the history of the sieve process. [...] There is a lot to do. A reasonable man, like myself, wouldn't spend 12% of his time, maybe, worrying about building sieves, if there wasn't any real use for them. It's very esoteric, of course, and since I am practically the only man working in this field you can see how widespread the interest in it is." [Lehmer, 1980]

A quick tour through the ENIAC

A quick tour through the ENIAC. [Goldstine, 1946, Goldstine and Goldstine, 1946, Burks and Burks, 1981]



The units of the ENIAC.

- 20 accumulators
- a multiplier, a divider and square rooter
- a constant transmitter and 3 function tables (ENIAC's main memory storage units)
- one master programmer (a central programming unit)
- cycling unit
- initiating unit
- a card reader and a printer

Some general aspects.

- Two kinds of circuits: the *numerical* circuits for storing and processing electric signals representing numbers and *programming* circuits for controlling the communication between the different parts of the machine.
- All units had to be programmed locally, connected through program cables
- Synchronization: the central programming pulse (CPP) = one addition time = 1/5000 second.
- Each unit takes an integer number of addition times to complete its operation. If so programmed it emits a programming pulse after finishing the operation, activating the next (sub)routine.

The accumulator. The main arithmetic units.

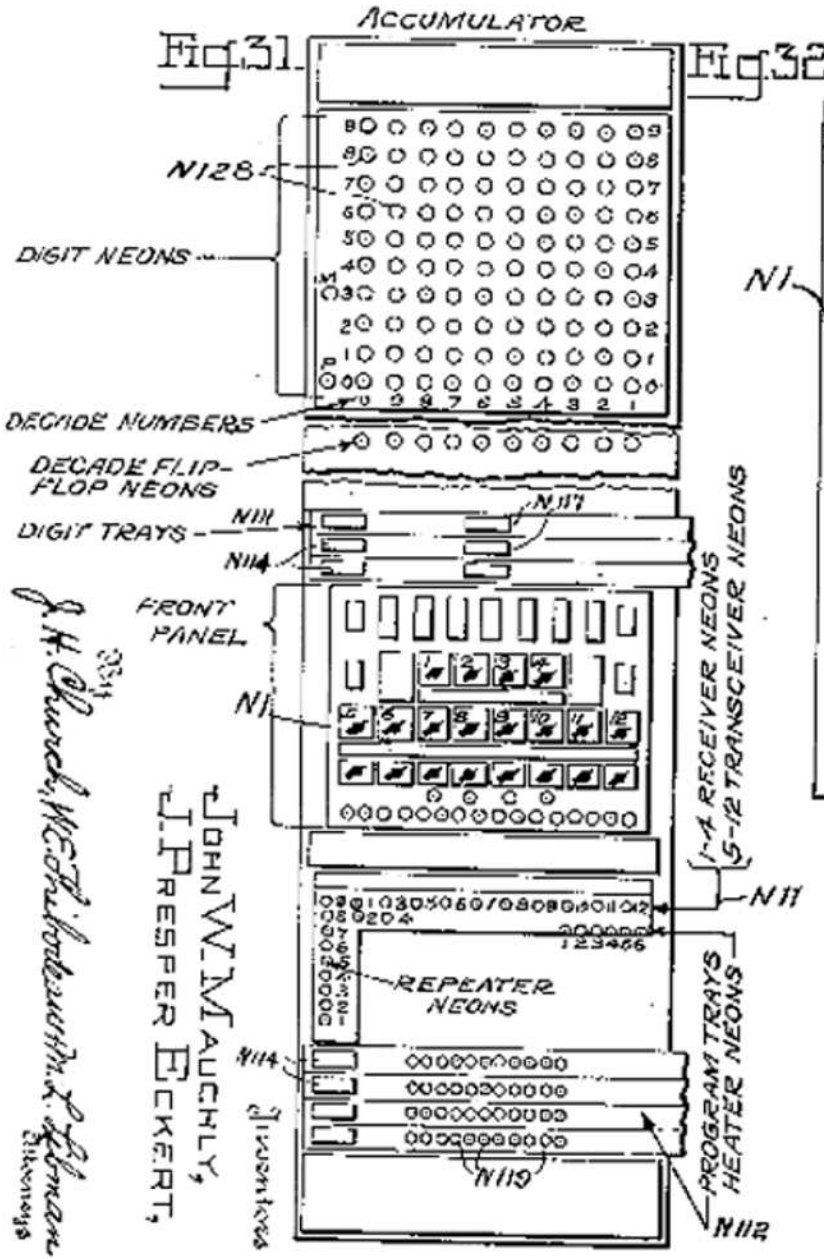
The numerical part

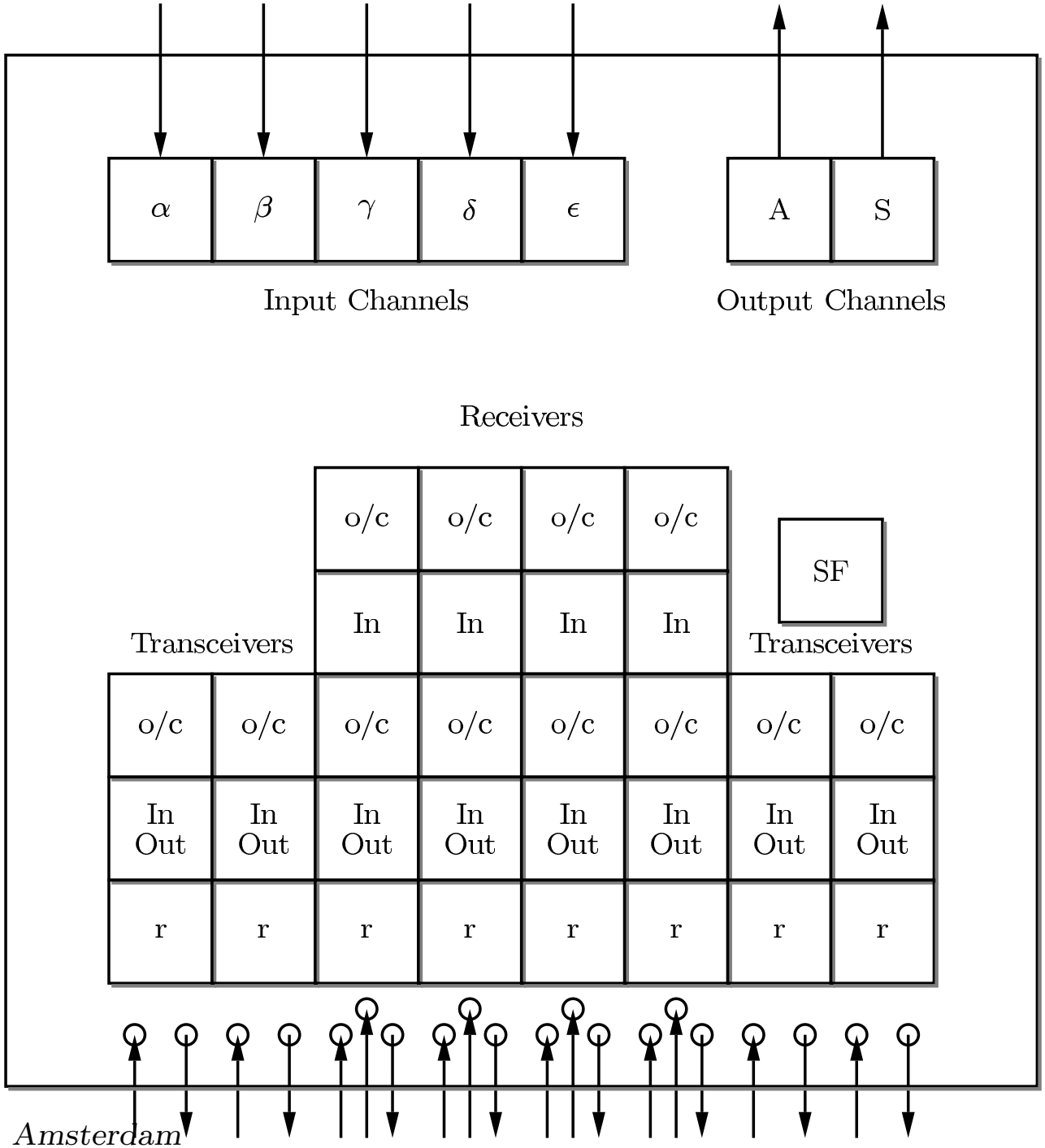
* Each can store a 10-decimal signed number in ten decade ring counters + PM-counter (for the sign)
* 5 input channels (α to ϵ), two output channels (A and S)to transmit a number n (through A) or its complement $10^{10} - n$ (through S)

The accumulator (Continued).

The programming part

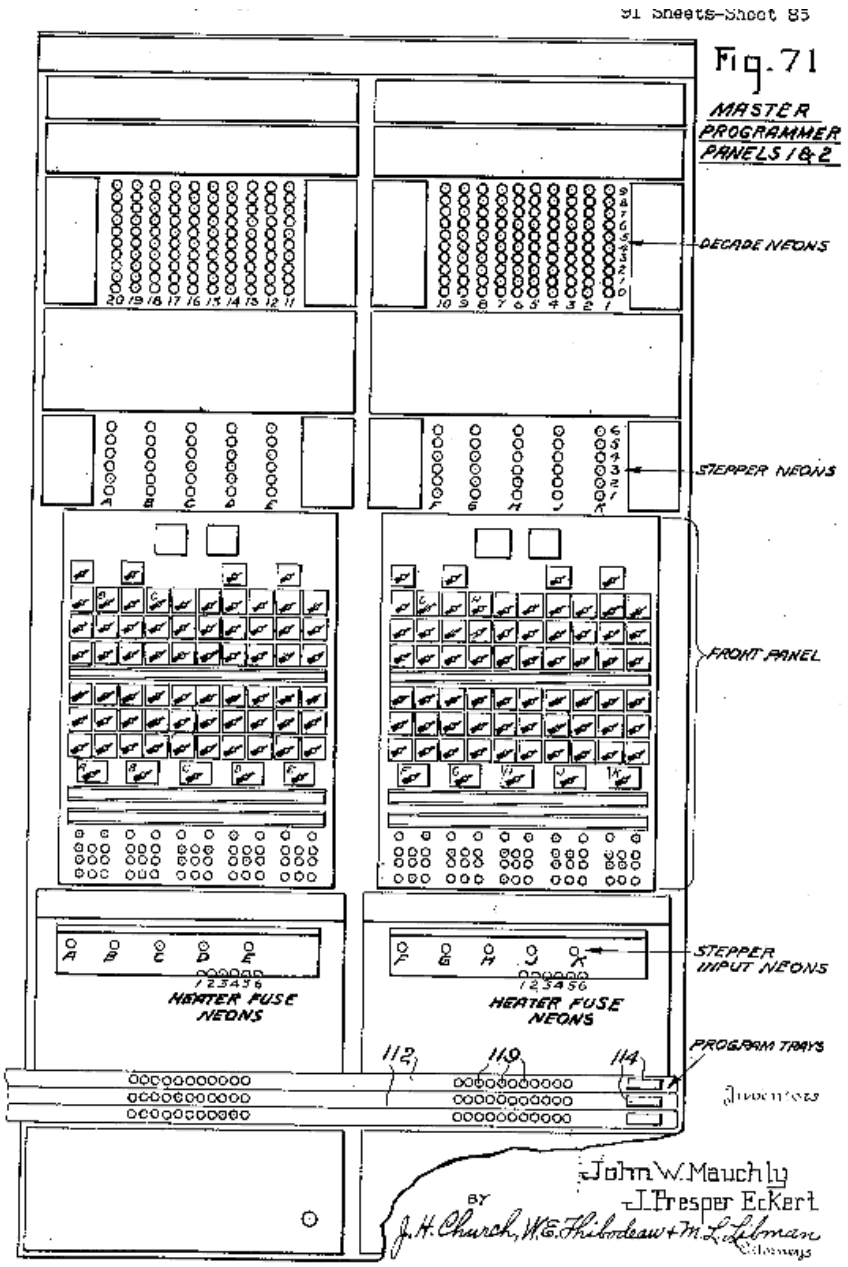
* 12 program controls: 4 receivers, 8 transceivers
* The transceiver: a program pulse input and output terminal, a clear-correct switch (to clear or not clear its content after a cycle; it could also be used to round off numerical results), an operation switch (to be set to α to ϵ, A, S, AS or 0, determining whether the accumulator should receive or transmit a number, or do nothing) and a repeat switch (with which it could either receive or transmit up to 9 times). (time = r, with $0 < r \leq 9$)
* The receiver: it has no program pulse output terminal and no repeater switch


ACCUMULATOR
Fig.31.
Fig.32
N128
DIGIT NEONS
N1
DECODE NUMBERS
DECADE FLIP-FLOP NEONS
DIGIT TRAYS
N111
N114
N117
FRONT PANEL
N1
1-4 RECEIVER NEONS
5-12 TRANSCEIVER NEONS
N11
REPEATER NEONS
N114
N119
PROGRAM TRAYS HEATER NEONS
N112
J. W. Mauchly, Philadelphia, Pa.
J. P. Eckert, Jr.
INVENTORS


α
β
γ
δ
ϵ
A
S
Input Channels
Output Channels
Receivers
o/c
In
SF
Transceivers
Transceivers
o/c
In
Out
r

The master programmer. Centralized programming memory.

- 10 independently functioning units, each having a 6-stage counter (called the stepper)
- 3 input terminals for each stepper counter (the stepper input, direct input and clear input)
- 6 output terminals for each stage of the stepper. Each such stage s was associated with a fixed number d_s by manually setting decade switches, and with 1 to 5 decade counters.


91 Sheets-Sheet 85
Fig. 71
MASTER PROGRAMMER PANELS 1 & 2
DECADE NEONS
STEPPER NEONS
FRONT PANEL
STEPPER INPUT NEONS
HEATER FUSE NEONS
HEATER FUSE NEONS
PROGRAM TRAYS
112
110
114
John W. Mauchly
J. Presper Eckert
BY
J. H. Church, W. E. Thilodeau + M. L. Libman

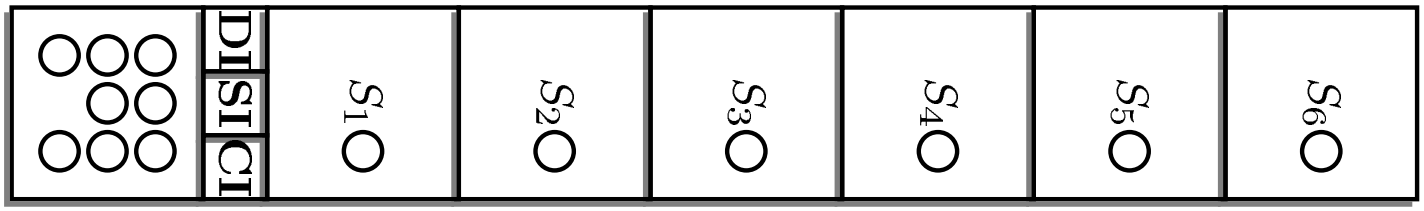


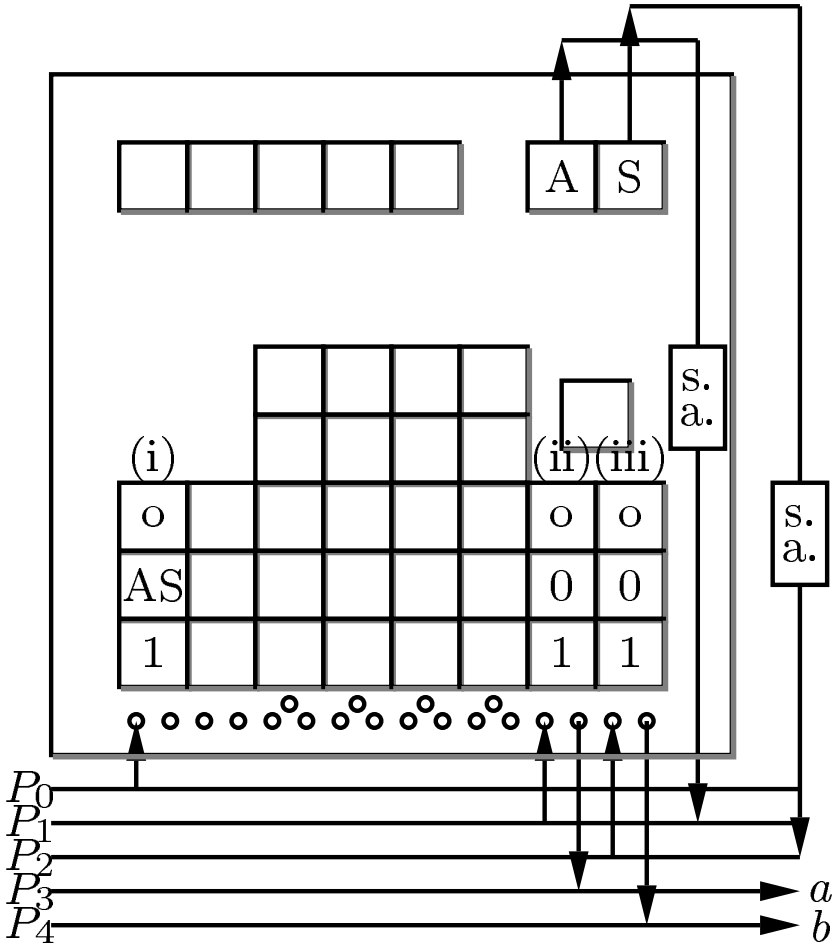
Figure 1: A Schematic (Reduced) Representation of a stepper counter of the Master Programmer.

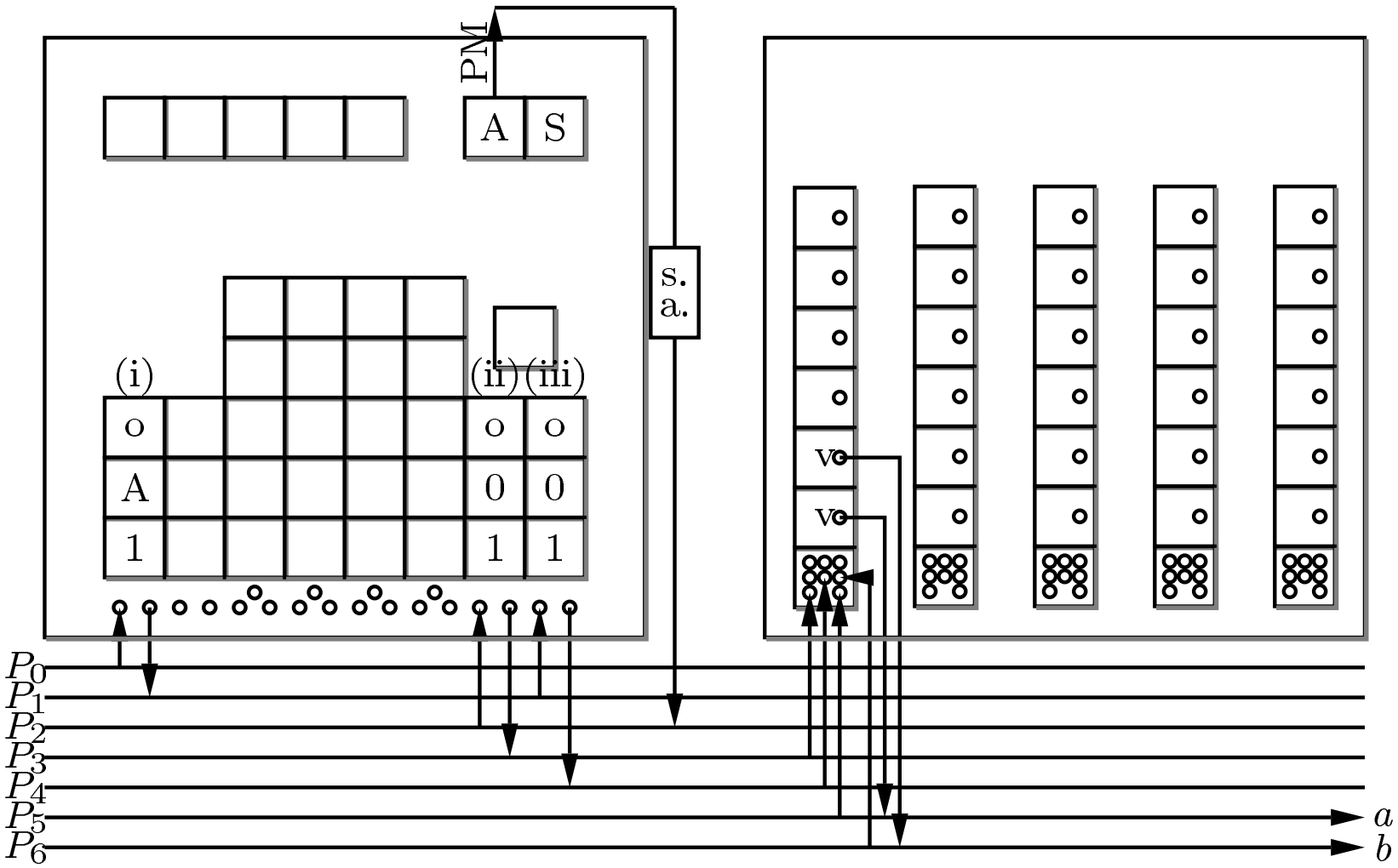
Branching...

- "magnitude discrimination" or "branching" : possible because 9 digit pulses were transmitted for sign indication M and none for sign indication P. The fact that digit pulses were transmitted for every digit except for 0 could be exploited in a similar manner.
- special adaptor for transforming digit pulse into programming pulse to the program pulse input terminal of an otherwise unused 'dummy (program) control'

Two methods

- 'IF' with two output channels of an accumulator
- 'IF' with one output channel and a stepper


A
S
s.
a.
(i)
(ii)
(iii)
o
o
o
AS
0
0
1
1
1
s.
a.
P_0
P_1
P_2
P_3
P_4
a
b


PM
A
S
s.
a.
(i)
(ii)
(iii)
o
A
1
o
o
0
0
1
1
v
v
P_0
P_1
P_2
P_3
P_4
P_5
P_6
a
b

The Lehmers' ENIAC program

A weekend off. Lehmer's ENIAC program

"*[Lehmer] had programmed the problem and run it on ENIAC, with J. Mauchly serving as "computer operator", during the three-day weekend of July 4, 1946. The running time of the problem occupied almost the entire weekend, around the clock, without a single interruption or malfunction. It was the most stringent performance test applied up to that time, and would be an impressive one even today. The problem was only a "test problem" from the point of view of the Army, but it provided an intrinsically important result in the theory of numbers.*"
[Alt, 1972]

"*[...] yes, an electronic computer could actually do an interesting problem in number theory – something as sophisticated in number theory – and produce useful results. There were many people who speculated about this – von Neumann among them – but to actually do it, to demonstrate it, was, I think, important to the post-war reputation of electronic computers among mathematicians.*"
[Akera, 2006]

The number-theoretical problem

- A special (but invalid) case of the converse of Fermat's little theorem

  Theorem 1 *If n divides $2^n - 2$ then n is a prime*

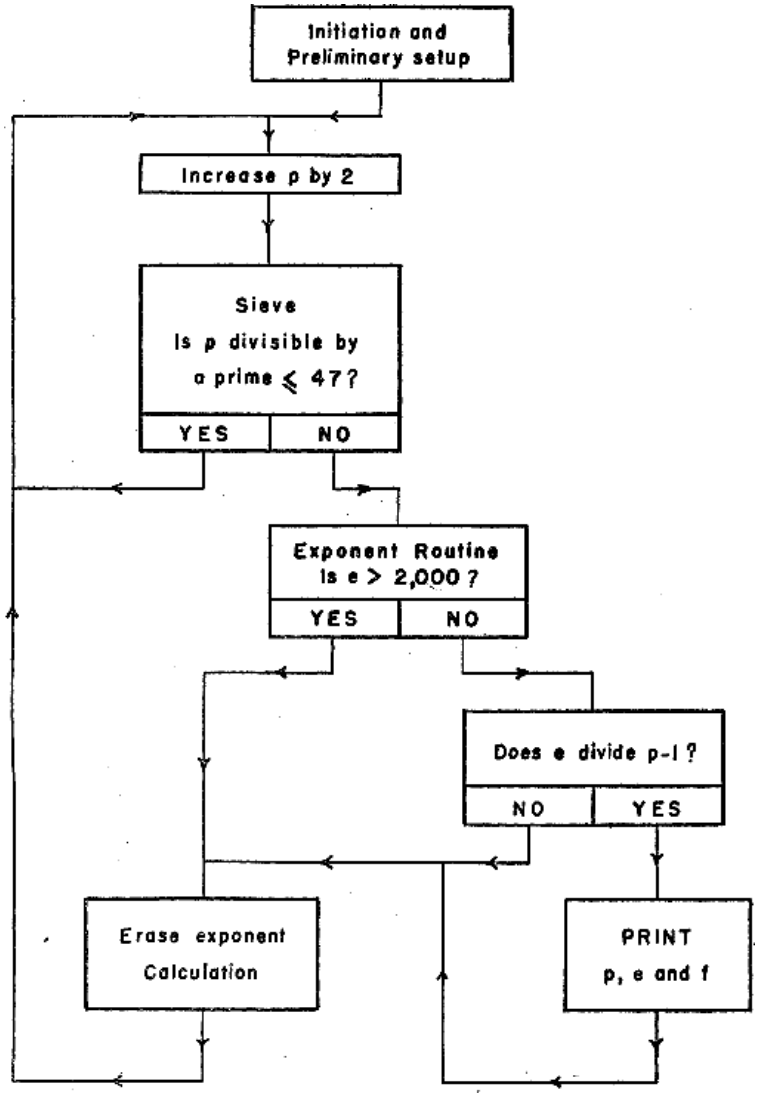
TABLE OF COMPOSITE SOLUTIONS n OF FERMAT'S CONGRUENCE $2^n \equiv 2 \pmod{n}$ AND THEIR SMALLEST PRIME FACTOR p

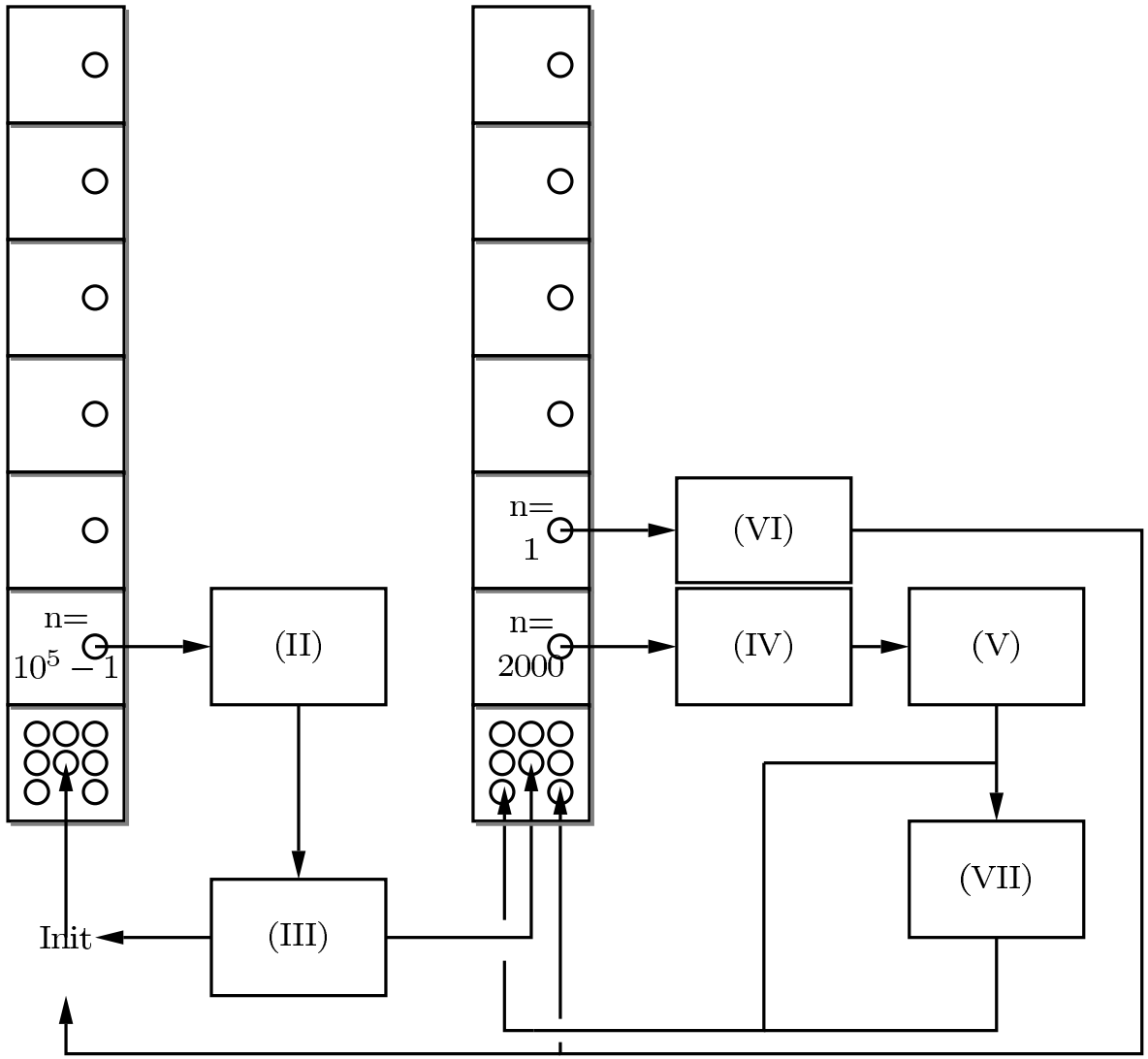
n	p	n	p	n	p
100463443	7577	312773	3541	558011	6449
618933	4729	413333	6067	940853	503
860997	9649	495083	1987	120296677	229
907047	5023	717861	1013	517021	2341
943201	5801	111202297	5273	838609	433
101152133	5807	370141	883	121062001	1201
158093	3673	654401	6101	128361	6961
218921	8713	112032001	4001	374241	6361
270251	9001	402981	3061	121472359	4409
276579	6163	828801	6133	122166307	739
954077	1597	844131	3067	396737	2857
102004421	2381	113359321	761	941981	$337 \cdot 491$
443749	4049	589601	$331 \cdot 571$	123330371	691
678031	3583	605201	7537	481777	3881
690677	2069	730481	433	559837	4177
690901	5851	892589	919	671671	9631
103022551	6121	114305441	6173	886003	1187
301633	7873	329881	7561	987793	709
104078857	6679	469073	3089	124071977	2089
233141	2441	701341	1229	145473	397
524421	5903	842677	2459	793521	4561
105007549	1033	115085701	1801	818601	2281
305443	2833	174681	773	125284141	4231
919633	4603	804501	5381	686241	6473
941851	1051	873801	1051	848577	2897
106485121	7297	116090081	6221	126132553	5023
622353	433	151661	7621	886447	6793
743073	1699	321617	5393	127050067	5347
107360641	2161	617289	2357	710563	9787
543333	4889	696161	2161	128027831	11161
108596953	7369	998669	1459	079409	5437
870961	2609	117246949	1597	124151	2311
109052113	4993	445987	5419	468957	2927
231229	2699	959221	2053	536561	8017
316593	3697	987841	7681	665319	2383
437751	5231	118466401	1249	987429	4637
541461	6043	119118121	2729	129205781	6563
879837	2707	204809	2383	256273	739
110135821	3967	261113	4657	461617	10177
139499	6427	378351	911	524669	2939

How was ENIAC used to compute composite numbers?

- The ENIAC was used to determine a list of exponents e of 2 mod p, i.e., the least value of n such that $2^n \equiv 1 \bmod p$, p prime and e is some divisor of $p-1$
- These exponents can be used to determine composite numbers of the form $2^{pq} - 2$ through the theorem:

  Theorem 2 *If p and q are odd distinct primes, then $2^{pq} - 2$ is divisible by pq if and only if p - 1 is divisible by the exponent to which 2 belongs modulo q and q - 1 is divisible by the exponent to which 2 belongs modulo p*
- A sieve was implemented on the ENIAC to determine primes relative to the first 15 primes, thus making use of the ENIAC's parallelism. The last prime p processed, after 111 hours of computing time, was $p = 4538791$


Initiation and
Preliminary setup
Increase p by 2
Sieve
Is p divisible by
a prime ⩽ 47?
YES
NO
Exponent Routine
Is e > 2,000?
YES
NO
Does e divide p-1?
NO
YES
Erase exponent
Calculation
PRINT
p, e and f


n=
$10^5 - 1$
(II)
(III)
Init
n=
1
n=
2000
(VI)
(IV)
(V)
(VII)

Computing the exponent e

"*The method used by the ENIAC to find the exponent of 2 modulo p differs greatly from the one used by human computer*" [Lehmer, 1949]

"*In contrast, the ENIAC was instructed to take an "idiot" approach, based directly on the definition of e, namely, to compute*

$$2^n \equiv \Gamma_n (\mathrm{mod} p), n = 1, 2, ...$$

until the value 1 appears or an until $n = 2001$*, whichever happens first. Of course, the procedure was done recursively by the algorithm:*

$$\Gamma_1 = 2, \Gamma_{n+1} = \begin{cases} \Gamma_n + \Gamma_n & \text{if } \Gamma_n + \Gamma_n < p \\ \Gamma_n + \Gamma_n - p & \text{otherwise} \end{cases}$$

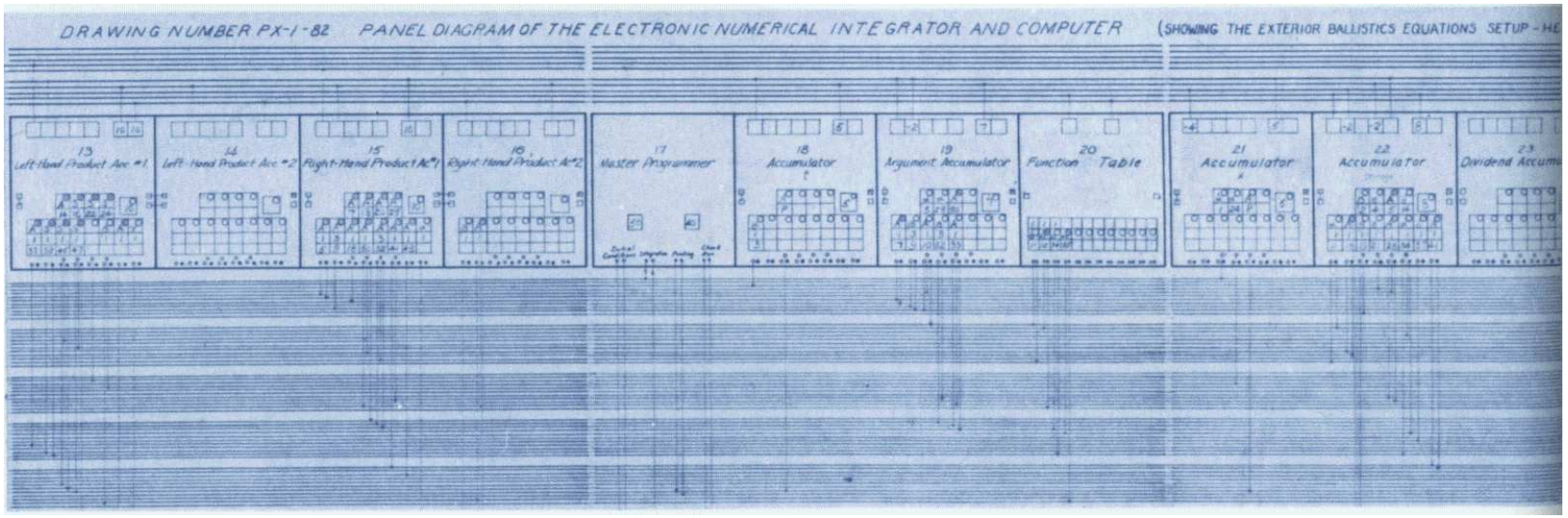
Only in the second case can Γ_{n+1} *be equal to 1. Hence this delicate exponential question in finding* $e(p)$ *can be handled with only one addition, subtraction, and discrimination at a time cost, practically independent of p, of about 2 seconds per prime. This is less time than it takes to copy down the value of p and in those days this was sensational.*" [Lehmer, 1974]

A Prime Sieve

- making use of the ENIAC's parallelism
- Minimizing the chance that $p = 2n + 1$ is not a prime relative to the primes ≤ 47.
- About 86 percent of the composites were thus eliminated after step 3 (sieve). The remaining 14 percent were required to pass a further test: namely $p - 1$ must be divisible by e (step 5). This requirement is so strict that the remaining number of composites is very small (25 out of 11336). Finally, these were eliminated by hand through comparison with D.N. Lehmer's list of primes.
- Eratosthenes's Sieve:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \dots \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \dots \end{pmatrix}$$

The Reconstruction.



Eniac set-up diagram.

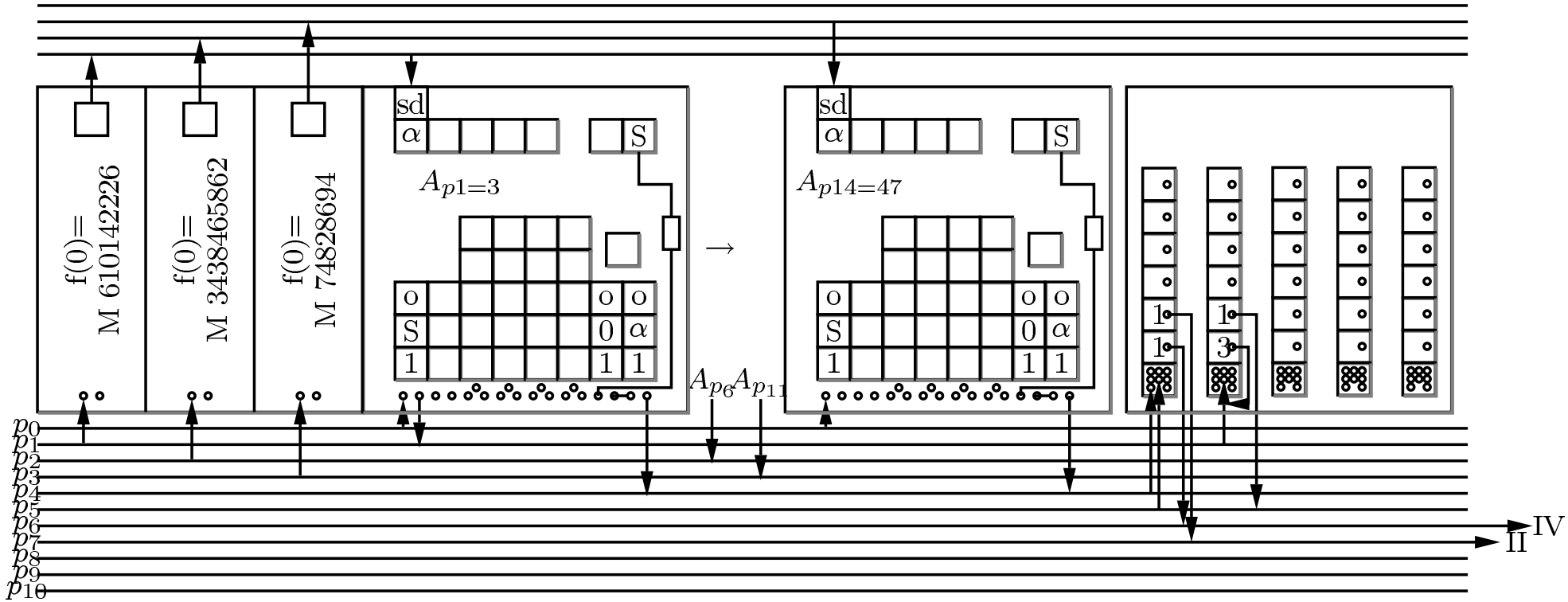
Reconstruction of the Sieve

- One accumulator for each prime $2 < p_j \leq 47$, resulting in 14 accumulators for the sieve.
- Initial set-up:
  * In each accumulator A_{p_j}, set complement of $p_j - 1$, e.g. $A_{p_{14}}$ will contain M 9999999954.
  * Initiating program pulse (pp) to (a) first transceiver T1 of each A_{p_j}, operation switch set to α, plus repeater set to 1 (b) the constant transmitter. This will send the number two to each of the A_{p_j}

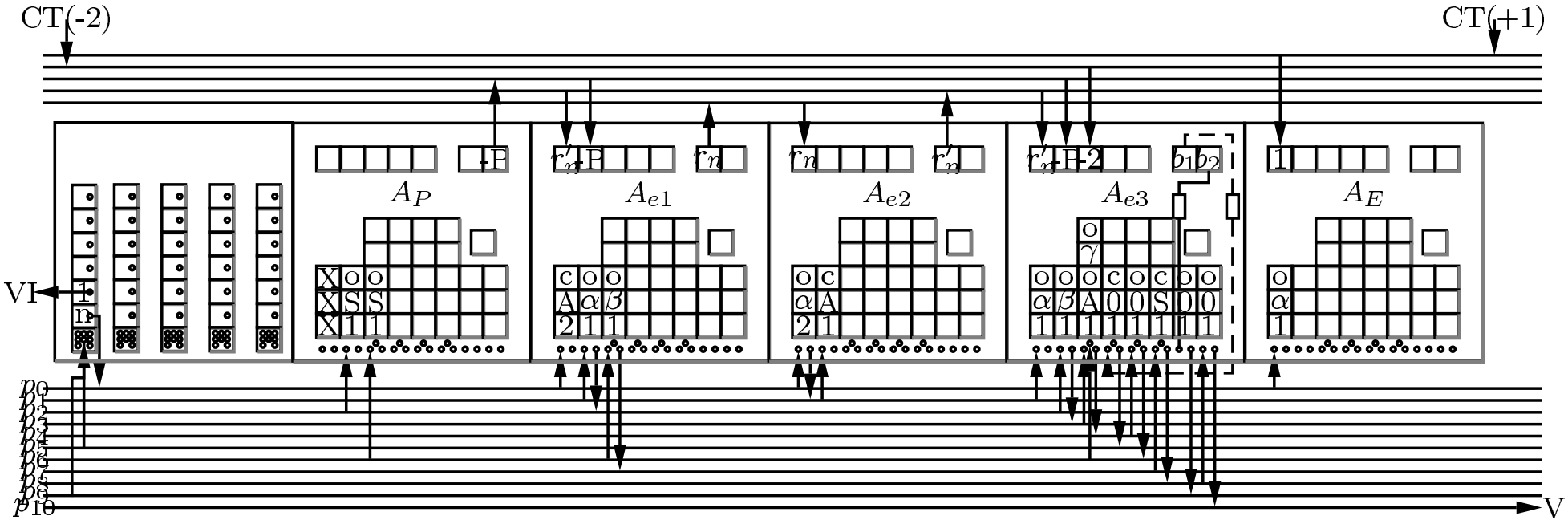
- The next steps: check for each A_{p_j} in parallel whether $P = 2r+1$ is divisible by p_j
  - **Checking routine**. Use of second branching method, connecting the PM lead of the S output of each of the A_{p_j} to 14 dummy controls (T2). If P is divisible by p_j, the number contained in A_{p_j} will be P 0000000000 and thus positive, while it will be negative in all other cases (this is why we use complements). If a given A_{p_j} stores P 0000000000, and P is thus divisible by p_j, A_{p_j} has to be reset to the complement of $2p_j$.
  - **The problem of loading** $2p_j$. Only those that contain P 0000000000 should receive a number (**Problem 1**) and each must receive a different number (**Problem 2**).

Problem 1. *Directly* connect the program pulse output terminal of each of the dummy controls (T2) of the A_{p_j} to the program pulse input terminal of one of the transceivers (T3) of each of the A_{p_j}. This could be done by using a *loaded program jumper* [Goldstine, 1946, 11.6.1]. Each T3 of an A_{p_j} is set to receive once through input channel α, β or γ depending on the group A_{p_j} belongs to.

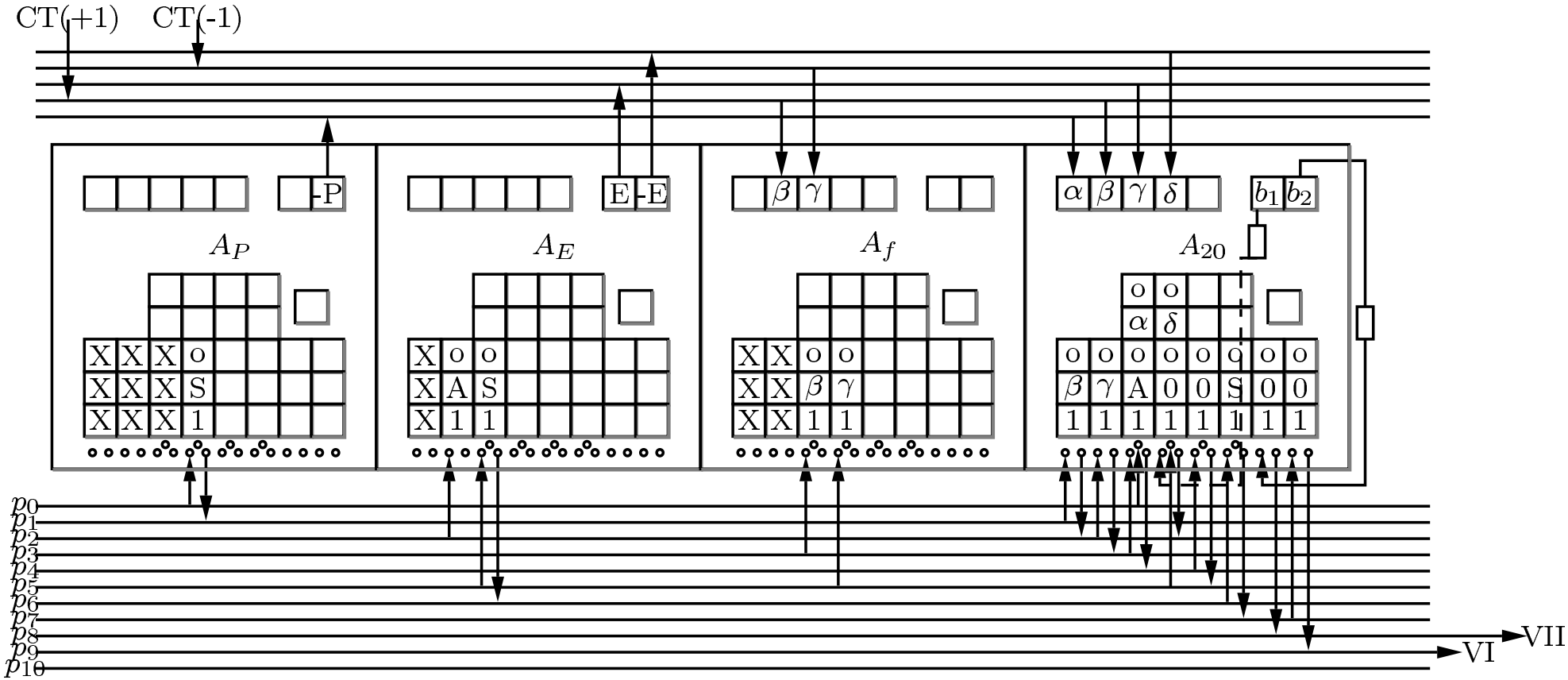
Problem 2. Use of the three function tables and special digit adaptors. The 14 A_{p_j}'s are divided into three groups: A_{p_1} – A_{p_5}, A_{p_6} – $A_{p_{10}}$, $A_{p_{11}}$ – $A_{p_{14}}$. In each group, the PP output terminal of T1 of rsp. A_{p_1}, A_{p_6} and $A_{p_{11}}$ is connected to three different program cables. The first of these cables sends a PP to function table 1, the second to function table 2 and the last to function table 3. Each of the function tables contains rsp. one of the following values: M 610142226, M 3438465862 and M 64828694 at place 0 (function value $f(0)$). Each of these values will be sent through the respective input channels α, β and γ and then be converted in the correct way through an adaptor connecting a shifter and deleter adaptors.


f(0)=
M 610142226
f(0)=
M 3438465862
f(0)=
M 74828694
sd
α
S
$A_{p1=3}$
o
S
1
o o
0 α
1 1
→
sd
α
S
$A_{p14=47}$
o
S
1
o o
0 α
1 1
1
1
1
3
$A_{p_6}A_{p_{11}}$
p_0
p_1
p_2
p_3
p_4
p_5
p_6
p_7
p_8
p_9
p_{10}
IV
II

Reconstruction of the Exponent Routine



Reconstruction of the Division Routine



Outline of the complete program

Step 1. Initiation and preliminary set-up, go to Step 2. Set A_{p_j} and function tables. Set numbers -2, + 2 and -1 on the CT (manually). Set A_P storing $2r+1$ to 1. Activation of the 14 A_{p_j}, A_P, A_e and the constant transmitter.

Step 2. Increase P by 2, goto Step 3.

Step 3. Sieve on p: Is p divisible by a prime ≤ 47? Y/N, goto Step 2/Step 4.

Step 4. Exponent routine to find e. Is $e > 2,000$? Yes/No, goto Step 7/Step 5. Use of 4 accumulators, i.e., $A_{e_1}, A_{e_2}, A_{e_3}, A_P, A_E$; three stepper counters (for checking rsp. $k > 2000, 2r_k - p > 0, 2r_k - p - 2 < 0$)

Step 5. Does e divide $p-1$? Yes/No, goto Step 6/Step 7. Use of 5 accumulators, including $A_P, A_E, A_{e_1}, A_{20}$. A_{20} receives P from A_P and next -1 from the CT. At the end of the computation, A_{e_1} will contain f, the number of times e can be subtracted from $p-1$

Step 6. Punch p, e and f $(p-1=ef)$, goto Step 7. One way is to use for A_P, A_{e_1} and A_E accumulators for which there is a static output to the printer.

Step 7. Erase exponent calculation, goto Step 2. Use the selective clear switch for the accumulators involved (except for A_P)

Conclusion

In fact, the programmer is a kind of engineer.
[Hopper and Mauchly, 1953, p. 1250]

- The reconstruction of historically important computer programs as a historiographic experiment
- Appreciating the difference between algorithms performed by human beings or those executed by a machine, through the interaction of a human being and the physical machine
- Significance of studying and reconstructing early computer programs to:
  * Clarify the development of programming techniques and computational methods in correlation with the development of the hardware.
  * Understand the evolution of the interaction and interface between the operator/programmer and the computer.
  * Because it's fun!

References

[Alt, 1972] Alt, F. L. (1972). Archaeology of computers – reminiscences, 1945–1947. *Communications of the ACM*, 15(7):693–694.

[Burks and Burks, 1981] Burks, A. W. and Burks, A. R. (1981). The eniac: First general-purpose electronic computer. *IEEE Annals of the History of Computing*, 3(4):310–399.

[Fritz, 1994] Fritz, W. B. (1994). Eniac – a problem solver. *Annals of the History of Computing IEEE*, 16(1):25–45.

[Goldstine, 1946] Goldstine, A. K. (1946). Report on the eniac, technical report i. Technical report, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia. Published in 2 vols.

[Goldstine and Goldstine, 1946] Goldstine, H. and Goldstine, A. (1946). The electronic numerical integrator and computer (eniac). *Mathematical Tabels and Other Aids to Computation*, 2(15):97–110.

[Hopper and Mauchly, 1953] Hopper, G. M. and Mauchly, J. (1953). Influence of programming techniques on the design of computers. *Proceedings of the IRE*, pages 1250–1254.

[Howlett et al., 1980] Howlett, J., Metropolis, N., and Rota, G.-C., editors (1980). *A History of Computing in the Twentieth Century*. Academia Press, New York. Proceeding of the International Research Conference on the History of Computing, Los Alamos, 1976.

[Lehmer, 1949] Lehmer, D. H. (1949). On the converse of fermat's theorem ii. *American Mathematical Monthly*, 56(5):300–309.

[Lehmer, 1974] Lehmer, D. H. (1974). The influence of computing on research in number theory. In LaSalle, J. P., editor, *The Influence of Computing on Mathematical Research and Education*, volume 20 of *Proceedings of Symposia in Applied Mathematics*, pages 3–12.

[Lehmer, 1980] Lehmer, D. H. (1980). A history of the sieve process. in: [Howlett et al., 1980], 445–456.